

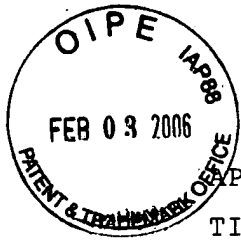


APPLICATION NO. 09/846,410

TITLE OF INVENTION: Multiple Data Rate Hybrid Walsh Codes for  
CDMA

INVENTOR: Urbain A. von der Embse

Clean copy of claims



APPLICATION NO. 09/846.410

TITLE OF INVENTION: Multiple Data Rate Hybrid Walsh Codes  
for CDMA

INVENTOR: Urbain A. von der Embse

## CLAIMS

WHAT IS CLAIMED IS:

Claim 1. (cancelled)

Claim 2. (cancelled)

Claim 3. (cancelled)

Claim 4. (cancelled)

Claim 5. (currently amended) A method for the generation of hybrid Walsh complex orthogonal codes for CDMA and for the plurality of other applications, said method comprising:

means for deriving the inphase permutation or reordering of the Walsh or Hadamard codes which places them in sequency correspondence which is the rate of phase rotation correspondence with the frequency of the discrete Fourier transform DFT codes and in even code correspondence with the real component codes of said DFT codes,

means for deriving the quadrature reordering of said Walsh or said Hadamard codes which places them in said sequency correspondence with the frequency of said DFT codes and in odd code correspondence with the quadrature component codes of said DFT,

means for using said inphase reordering to generate the inphase component codes of said hybrid Walsh codes,  
and

means for using said quadrature reordering to generate the quadrature component codes of said hybrid Walsh codes.

Claim 6. (currently amended) A method for the generation of generalized hybrid Walsh orthogonal codes for CDMA and for the plurality of other applications, from code sets which include said hybrid Walsh, said Hadamard, said Walsh, said DFT, pseudo-noise PN, and the plurality of codes, said method comprising:  
means for generating said codes using tensor product techniques for codes selected from the plurality of said code sets,  
means for generating said codes using direct product techniques for codes selected from the plurality of said code sets,  
means for generating said codes using functional combining techniques for codes selected from the plurality of said code sets, and  
means for generating said codes using combinations of said tensor product techniques, said direct product techniques, and said functional combining techniques for codes selected from the plurality of said code sets.

Claim 7. (new) For said codes in claims 1,2 a method for mapping multiple data rate user symbols onto the code vectors, said method comprising:  
means for supporting multiple data rate users with symbol rates from the menu  $1/NT, 2/NT, \dots, 2/T$  for a N chip said code with a T second chip interval,  
means for assigning said users with said like symbol rates to the groups  $u_{M-1}, u_{M-2}, \dots, u_1, u_0$  for the respective said menu symbol rates  $1/NT, 2/NT, \dots, 2/T$ ,  
means for assigning code indices from the binary indexing fields  $d_{M-1}, d_{M-2}d_{M-1}, \dots, d_1d_2 \dots d_{M-2}d_{M-1}, d_0d_1d_2 \dots d_{M-2}d_{M-1}$  to said users in said respective groups  $u_0, u_1, \dots, u_{M-2}$ ,  
means for generating the code index  $d = d_0 + 2d_1 + 4d_2 + \dots + (N/2)d_{M-1} = 0, 1, 2, \dots, N-1$  and for mapping said index onto the N code vectors, and

means for assigning said users to said code vectors using said user index fields and said mapping of said code index onto said code vectors.

**Claim 8.** (new) Said codes in claims 5,6 have fast encoding algorithms, wherein a representative fast algorithm for said codes in claim 5 comprising:

means for using said index fields in claim 7 to arrange the input data symbol set in the format  $Z(d_0, d_1, \dots, d_{M-2}, d_{M-1})$ ,

means for performing pass 1 of said fast encoding algorithm by multiplying said  $Z$  by the kernel  $[(-1)^{dr_0 n_{M-1} + j} (-1)^{di_0 n_{M-1}}]$  and summing over  $dr_0, di_0 = 0, 1$  to yield the partially encoded symbol set  $Z(n_{M-1}, d_1, \dots, d_{M-2}, d_{M-1})$  where  $dr_0 = cr(d_0)$  and  $cr(d)$  is said real axis code for  $d$ ,  $di_0 = ci(d_0)$  where  $ci(d)$  is said imaginary axis code for  $d$ , and  $n_{M-1}$  is a binary code chip coefficient in the code chip indexing  $n = n_0 + 2n_1 + \dots, (N/4)n_{M-2} + (N/2)n_{M-1}$ ,

means for performing passes  $m=2, 3, \dots, M-1$  of said fast encoding algorithm by multiplying

$Z(n_{M-1}, n_{M-2}, \dots, n_{M-m+1}, d_{m-1}, \dots, d_{M-2}, d_{M-1})$  by the kernel  $[(-1)^{dr_{m-1}(n_{M-m} + n_{M-m+1}) + j} (-1)^{di_{m-1}(n_{M-m} + n_{M-m+1})}]$  and summing over  $dr_{m-1}, di_{m-1} = 0, 1$  to yield the partially encoded symbol set  $Z(n_{M-1}, n_{M-1}, n_{M-2}, \dots, n_{M-m}, d_m, \dots, d_{M-2}, d_{M-1})$ ,

means for performing pass  $M$  of said fast encoding algorithm by multiplying  $Z(n_{M-1}, n_{M-2}, \dots, n_2, n_1, d_{M-1})$  by the kernel  $[(-1)^{dr_{M-1}(n_0 + n_1) + j} (-1)^{di_{M-1}(n_0 + n_1)}]$  and summing over  $dr_{M-1}, di_{M-1} = 0, 1$  to yield the encoded symbol set

$Z(n_{M-1}, n_{M-1}, n_{M-2}, \dots, n_2, n_1, n_0)$ , and

means for reordering the encoded symbol set in the ordered output format  $Z(n_0, n_1, \dots, n_{M-2}, n_{M-1})$ .

Claim 9. (new) Said codes in claims 5,6 have fast decoding algorithms, wherein a representative fast algorithm for said codes in claim 5 comprising:

means for performing pass 1 of said fast decoding algorithm by multiplying said  $Z(n_0, n_1, \dots, n_{M-2}, n_{M-1})$  from claim 8 by the kernel  $[(-1)^{n_0} dr_{M-1} + j(-1)^{n_0} di_{M-1}]$  and summing over  $n_0=0,1$  to yield the partially decoded symbol set

$$Z(d_{M-1}, n_1, \dots, n_{M-2}, n_{M-1}),$$

means for performing passes  $m=2,3,\dots,M-1$  of said fast encoding algorithm by multiplying

$Z(d_{M-1}, d_{M-2}, \dots, d_{M-m+1}, n_{m-1}, \dots, n_{M-2}, n_{M-1})$  by the kernel  $[(-1)^{n_{m-1}}(dr_{M-m} + dr_{M-m+1}) + j(-1)^{n_{m-1}}(di_{M-m} + di_{M-m+1})]$  and summing over  $n_{m-1}=0,1$  to yield the partially encoded symbol set

$$Z(d_{M-1}, d_{M-1}, d_{M-2}, \dots, d_{M-m}, n_m, \dots, n_{M-2}, n_{M-1}),$$

means for performing pass M of said fast encoding algorithm by multiplying  $Z(d_{M-1}, d_{M-2}, \dots, d_2, d_1, n_{M-1})$  by the kernel  $[(-1)^{n_{M-1}}(dr_0 + dr_1) + j(-1)^{n_{M-1}}(di_0 + di_1)]$  and summing over  $n_{M-1}=0,1$  to yield the encoded symbol set

$$Z(d_{M-1}, d_{M-1}, d_{M-2}, \dots, d_2, d_1, d_0), \text{ and}$$

means for reordering the decoded symbol set in the ordered output format  $Z(d_0, d_1, \dots, d_{M-2}, d_{M-1})$ .